Appendix

MAP Decoding in Channels with Memory

William Turin, *Senior Member, IEEE*

Abstract

The expectation-maximization (EM) algorithm is popular in estimating parameters of various statistical models. In this paper, we consider applications of the EM algorithm to the maximum a posteriori (MAP) sequence decoding assuming that sources and channels are described by hidden Markov models (HMMs). HMMs can accurately approximate a large variety of communication channels with memory and, in particular, wireless fading channels with noise. The direct maximization of the a posteriori probability (APP) is too complex. The EM algorithm allows us to obtain the MAP sequence estimation iteratively. Since each step of the EM algorithm increases the APP, the algorithm can improve performance of any decoding procedure.

Keywords: Maximum a posteriori decoding, EM algorithm, hidden Markov model, fading channel

W. Turin is with the Communications Research Department, AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ 07932 (e-mail: wt@research.att.com).

1. INTRODUCTION

Maximum a posteriori (MAP) sequence decoding is optimum, because it minimizes the probability of error. [27] It is usually performed by the Viterbi algorithm and most recently by the turbo decoding algorithm for a special class of codes. These algorithms are directly applicable if communications channels are memoryless. However, they perform an approximate MAP decoding if channel errors are bursty which is the case in wireless communications due to fading. It is usually very difficult to find a sequence MAP estimate directly for channels with memory. The expectation-maximization (EM) algorithm has been successfully applied by Georghiades and Han, [8] Zeger and Kobayashi, [28] and Georghiades [9] to decoding information transmitted over the fading wireless channel. In these papers, the fading channel is modeled by a complex Gaussian process.

Alternatively, the wireless channel can be modeled by hidden Markov models (HMMs). [22,23] It can be shown that HMMs are general enough to approximate not only fading, but also other types of signal distortion such as interference and non-Gaussian noise. [23] In this paper, we demonstrate that the EM algorithm can be applied to MAP decoding if channel errors are described by an HMM. In our approach, the signal parameters are obtained by applying the EM algorithm to maximizing the a posteriori probability (APP) of the transmitted symbols.

In developing the MAP decoding algorithm it is usually assumed that the communication channel is memoryless which is achieved by interleaving. However, in the majority of real channels the error dependence extends over long time intervals. Complete independence is impossible to achieve due to the information delivery delay constraints and system memory limitations. We should also remember that a memoryless channel has lower capacity than a channel with memory with the same bit-error rate. [7,10] Therefore, it is important to develop decoding algorithms for channels with memory.

There are many different models of channels with memory which correspond to various channel impairments such as fading, interference, and noise. All of them can be accurately approximated by hidden Markov models (HMMs). It can be shown that HMMs represent a dense family allowing us to approximate a large variety of stochastic processes. Their application in

many diverse fields (such as speech, image, and handwriting recognition, experimental genetics, sociology, stock market modeling, etc) serves as experimental evidence of their generality. Another reason for their popularity is the relative simplicity of their use.

In this paper we develop the MAP decoding algorithm for a general input-output HMM (IOHMM) which, as we will see, incorporates source and channel HMMs. For the special class of models, the decoding can be performed using the Viterbi algorithm. However, in the general case this algorithm gives only an approximate solution. We demonstrate that this approximate solution can be improved by the expectation maximization (EM) algorithm. The paper is organized as follows. In section 2, we discuss the relationship between various decoding criteria. In section 3, we consider the IOHMMs and their application to computing the transmitted symbol sequence APP. In section 4, we develop the MAP EM decoding algorithm and illustrate its application to decoding of block and convolutional codes.

2. MAP DECODING

Suppose that we have an input-output system whose input is described by a sequence of symbols $X_1^T = (X_1, X_2, \ldots, X_T)$ and the corresponding output is $Y_1^T = (Y_1, Y_2, \ldots, Y_T)$. Our goal is to choose the most probable input which produced the observed output Y_1^T.

The optimal estimator that maximizes the probability of X_1^T correct decoding is the maximum a posteriori (MAP) estimator: [27]

$$\hat{X}_1^T = \underset{X_1^T}{argmax}\, Pr(X_1^T \mid Y_1^T) \tag{1}$$

where $Pr(\cdot)$ denotes the corresponding probability or probability density function. Since the maximization does not depend on Y_1^T, the MAP estimate can be obtained by maximizing the unnormalized APP

$$\hat{X}_1^T = \underset{X_1^T}{argmax}\, Pr(X_1^T, Y_1^T) = \underset{X_1^T}{argmax}\, Pr(Y_1^T \mid X_1^T) Pr(X_1^T). \tag{2}$$

It follows from this equation that the maximum likelihood (ML) and MAP estimates coincide if the input is uniformly distributed. This assumption is often made when there is no information about the input probability distribution. [27] However, better results can be achieved if we

exploit the input (source) statistics. Since the ML estimate can be viewed as a special case of the MAP estimate, we consider only the latter in the sequel.

It is often necessary to estimate a subset of the input sequence and in particular a single symbol:

$$\hat{X}_t = \underset{X_t}{argmax}\ Pr(X_t, Y_1^T). \tag{3}$$

In many applications the estimation must be performed before receiving the whole sequence Y_1^T. If estimation of X_t is made on the basis of $Y_1^{t+\tau}$ with $\tau > 0$, it is called a *fixed lag smoothing*; if $\tau = 0$, it is called a *filtering*; if $\tau < 0$, it is called a *prediction*. To solve the previous equations we need to develop algorithms for calculating the corresponding probability measures and their maximization. This is done by modeling the input-output system.

3. HIDDEN MARKOV MODELS

An input-output HMM $\lambda = (S, X, Y, \boldsymbol{\pi}, \{\mathbf{P}(x,y)\})$ is defined by its internal states $S = \{1, 2, \ldots, n\}$, inputs X, outputs Y, initial state probability vector $\boldsymbol{\pi}$, and the input-output probability density matrices (PDM) $\mathbf{P}(x,y)$, $x \in X$, $y \in Y$ whose elements $p_{ij}(x,y) = Pr(j,x,y \mid i)$ are the conditional probability density functions (PDF) of input x and corresponding output y after transferring from state i to state j. It is assumed that the state sequence $S_0^t = (S_0, S_1, \ldots, S_t)$, input sequence $X_1^t = (X_1, X_2, \ldots, X_t)$, and output sequence $Y_1^t = (Y_1, Y_2, \ldots, Y_t)$ possess the following Markovian property

$$Pr(S_t, X_t, Y_t \mid S_0^{t-1}, X_1^{t-1}, Y_1^{t-1}) = Pr(S_t, X_t, Y_t \mid S_{t-1}).$$

According to this model, the PDF of the input sequence X_1^T and output sequence Y_1^T has the form [23]

$$p_T(X_1^T, Y_1^T) = \boldsymbol{\pi} \prod_{i=1}^{T} \mathbf{P}(X_i, Y_i)\ \mathbf{1} \tag{4}$$

where $\mathbf{1}$ is a column vector of n ones.

If the source sequence is modeled by an autonomous HMM with states $S_t^{(s)}$ and PDM $\mathbf{P}_s(x) = [Pr(S_t^{(s)}, x \mid S_{t-1}^{(s)})]_{n_s n_s}$ and, for each input sequence, the channel is modeled by the

conditional HMM with states $S_t^{(c)}$ and PDM $\mathbf{P}_c(y|x) = [Pr(S_t^{(c)}, y|S_{t-1}^{(c)}, x)]_{n_c, n_c}$, then the combined IOHMM is described by the PDM of the form

$$\mathbf{P}(X,Y) = \mathbf{P}_s(X) \otimes \mathbf{P}_c(Y|X) \tag{5}$$

where $\otimes$ denotes the matrix Kronecker product. In other words, the combined model is an HMM whose states represent all possible combinations of the transmitted sequence states and channel states.

In the most popular model, the conditional output PDM has the form

$$\mathbf{P}_c(Y|X) = \mathbf{P}_c \mathbf{B}_c(Y|X) \tag{6}$$

where $\mathbf{P}_c = [p_{ij}]_{n_c, n_c}$ is the channel state transition probability matrix and $\mathbf{B}_c(Y|X)$ is a diagonal matrix of the state output probabilities. For example, according to the Gilbert-Elliott [3,10] model, the channel has two states: "good" and "bad". In the good state, errors occur with small probability b_1 while in the bad state they occur with larger probability b_2. If we assume that the first state is good and the second state is bad, then the model is described by equation (6) with

$$\mathbf{B}_c(X|X) = \begin{bmatrix} 1-b_1 & 0 \\ 0 & 1-b_2 \end{bmatrix} \quad \mathbf{B}_c(\bar{X}|X) = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \tag{7}$$

where $\bar{X}$ denotes the complement of X. The two-state model is the simplest HMM for channels with memory. Models with larger state space are often needed. [19,23] There are several HMMs for fading channels with additive white Gaussian noise (AWGN). In these models, the HMM states are usually associated with the channel fade levels while the state output conditional probability density functions are Gaussian [13,18,22,23]

$$b_k(Y_t|X_t) \propto \exp(-\parallel Y_t - a_k X_t \parallel^2 / N_0). \tag{8}$$

Let us consider now the transmitted sequence modeling by HMMs. Usually, the source HMM is obtained by fitting it to experimental data. This process is called the HMM training. [16] If the transmitted sequence is generated by a trellis coded modulator (TCM), [6,11,24] then it can be described as a finite state machine:

$$\begin{aligned} S_{t+1}^{(s)} &= f_t(S_t^{(s)}, I_t) \\ X_t &= g_t(S_t^{(s)}, I_t). \end{aligned} \tag{9}$$

After receiving an information symbol I_t in state $S_t^{(s)}$ the machine transfers to state $S_{t+1}^{(s)} = f_t(S_t^{(s)}, I_t)$ and outputs a symbol $X_t = g_t(S_t^{(s)}, I_t)$. The system is called TCM, because its state transition graph resembles a trellis whose nodes represent the states $S_t^{(s)}$ for $t = 1, 2, \ldots, T$ and edges represent all possible state transitions. In a typical implementation, the first equation in (9) represents a convolutional encoder while the second equation represents a modulator.

It follows from this description that the modulator output symbols are not independent, even if the source symbols are independent. From a statistical point of view, TCM can be described as a method of creating correlated sequences that are resistant to channel errors, allowing us to recover the source sequences with high reliability. It follows from the TCM definition that if the source symbols can be modeled by an HMM, the output process is also an HMM.

4. MAP SEQUENCE DECODING VIA THE EM ALGORITHM

If the channel is modeled by an IOHMM, the MAP estimate of the sequence X_1^T maximizes the right hand side of equation (4). However, as we can see, the direct maximization is a difficult problem. If X_t is a discrete variable, then we need to consider all possible sequences X_1^T. Thus, the complexity grows exponentially with T. In the special case, when the input sequence X_1^T is uniquely determined by the sequence of states S_1^T, the maximum can be found by the Viterbi algorithm. [4] In this paper, we show that the general problem can be solved iteratively using the EM algorithm which, in our case, is a combination of the forward-backward and Viterbi algorithms. [2,16] The EM algorithm converges monotonically, [2] therefore, even single step of the EM algorithm allows us to improve performance of any decoding algorithm.

To develop the algorithm, we define the complete data probability distribution as [23, Sec. 3.2.2]

$$\psi(z, X_1^T, Y_1^T) = \pi_{i_0} \prod_{t=1}^{T} p_{i_{t-1} i_t}(X_t, Y_t) \tag{10}$$

where $z = i_0^T$ is the HMM state sequence and $p_{ij}(X,Y)$ are the elements of the matrix $\mathbf{P}(X,Y)$. The MAP sequence estimate of equation (2) can be obtained iteratively by the following EM

algorithm [2] [23]

$$X_{1,p+1}^T = \arg\max_{X_1^T} Q(X_1^T, X_{1,p}^T), \quad p=0,1,2,... \tag{11}$$

where $Q(X_1^T, X_{1,p}^T)$ is the auxiliary function which, in our case, can be written as

$$Q(X_1^T, X_{1,p}^T) = \sum_z \psi(z, X_{1,p}^T, Y_1^T) \log \psi(z, X_1^T, Y_1^T).$$

Using the relationship between equations (10) and (4) we can rewrite the auxiliary function in the following form [23]

$$Q(X_1^T, X_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{t,ij}(X_{1,p}^T) \log p_{ij}(X_t, Y_t) + C \tag{12}$$

where C does not depend on X_1^T

$$\gamma_{t,ij}(X_{1,p}^T) = \alpha_i(X_{1,p}^{t-1}, Y_1^{t-1}) p_{ij}(X_{t,p}, Y_t) \beta_j(X_{t+1,p}^T, Y_{t+1}^T)$$

$\alpha_i(X_{1,p}^t, Y_1^t)$ and $\beta_i(X_{t+1,p}^T, Y_{t+1}^T)$ are the elements of the following forward and backward probability vectors

$$\boldsymbol{\alpha}(X_1^t, Y_1^t) = \boldsymbol{\pi} \prod_{i=1}^{t} \mathbf{P}(X_i, Y_i) \quad \text{and} \quad \boldsymbol{\beta}(X_1^t, Y_1^t) = \prod_{i=t}^{T} \mathbf{P}(X_i, Y_i)\mathbf{1}. \tag{13}$$

It follows from these equations that the forward probability vectors can be evaluated by the forward algorithm

$$\boldsymbol{\alpha}(X_1^0, Y_1^0) = \boldsymbol{\pi}, \quad \boldsymbol{\alpha}(X_1^t, Y_1^t) = \boldsymbol{\alpha}(X_1^{t-1}, Y_1^{t-1}) \mathbf{P}(X_t, Y_t) \tag{14}$$

and the backward probability vectors can be evaluated by the backward algorithm

$$\boldsymbol{\beta}(X_{T+1}^T, Y_{T+1}^T) = \mathbf{1}, \quad \boldsymbol{\beta}(X_t^T, Y_t^T) = \mathbf{P}(X_t, Y_t) \boldsymbol{\beta}(X_{t+1}^T, Y_{t+1}^T). \tag{15}$$

As we can see, the auxiliary function $Q(X_1^T, X_{1,p}^T)$ is much simpler than $p_T(X_1^T, Y_1^T)$. In the majority of practical cases it is not difficult to find a maximum of the auxiliary function. For example, if source coded speech signal, which is modeled by an HMM with the mixture of Gaussian state-conditional PDFs, is transmitted over the slowly fading channel with additive Gaussian noise, it is possible to find the closed form solution of equation (11). [14] However, the matrix $\mathbf{P}(X_t, Y_t)$ size might be large. The previous equations dimensionality can be reduced if this matrix has a special form.

5. MAP DECODING OF TCM SIGNALS ON CHANNELS WITH MEMORY

For a TCM system with an i.i.d information sequence, equation (5) takes the form

$$\mathbf{P}(X_t, Y_t) = [p_{S_t^{(s)} S_{t+1}^{(s)}} \mathbf{P}_c(Y_t \mid X_t)] \tag{16}$$

where

$$p_{S_t^{(s)} S_{t+1}^{(s)}} = \begin{cases} Pr(I_t) & \text{if} \quad S_{t+1}^{(s)} = f_t(S_t^{(s)}, I_t) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

which means that the matrix $\mathbf{P}(X_t, Y_t)$ is sparse. In this case, we can write

$$p_T(I_1^T, Y_1^T) = \boldsymbol{\pi}_c \prod_{t=1}^{T} p_{S_t^{(s)} S_{t+1}^{(s)}} \mathbf{P}_c(Y_t \mid X_t) \mathbf{1}.$$

Here and in the following equations, $\boldsymbol{\pi}_c$ is a vector of the initial probabilities of the channel states, $X_t = g_t(S_t^{(s)}, I_t)$, and the product is taken along the state trajectory $S_{t+1}^{(s)} = f_t(S_t^{(s)}, I_t)$ for $t = 1, 2, \ldots, T$. Thus, for the TCM system, instead of a large sparse matrix $\mathbf{P}(X_t, Y_t)$, we can use smaller matrices $\mathbf{P}_c(Y_t \mid X_t)$ for computing the APP.

If we assume that all the information symbols are equiprobable, then the MAP estimate is equivalent to the ML estimate:

$$\hat{I}_1^T = \underset{I_1^T}{\arg\max}\, \boldsymbol{\pi}_c \prod_{t=1}^{T} \mathbf{P}_c(Y_t \mid X_t) \mathbf{1}.$$

The auxiliary function can be also written in terms of the smaller matrices:

$$Q(I_1^T, I_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \gamma_{t,ij}(I_{1,p}^T) \log p_{c,ij}(Y_t | X_t) + C \tag{18}$$

where $X_t = g_t(S_t^{(s)}, I_t)$ and

$$\gamma_{t,ij}(I_{1,p}^T) = \alpha_i(Y_1^{t-1} | I_{1,p}^{t-1}) p_{c,ij}(Y_t | X_{t,p}) \beta_j(Y_{t+1}^T | I_{t+1,p}^T) \tag{19}$$

$\alpha_i(Y_1^{t-1} | I_{1,p}^{t-1})$ and $\beta_j(Y_{t+1}^T | I_{t+1,p}^T)$ are the elements of the forward and backward probability vectors

$$\alpha(Y_1^t | I_1^t) = \boldsymbol{\pi}_c \prod_{i=1}^{t} \mathbf{P}_c(Y_i | X_i) \quad \text{and} \quad \beta(Y_1^t | I_1^t) = \prod_{i=t}^{T} \mathbf{P}_c(Y_i | X_i) \mathbf{1} \tag{20}$$

which can be computed by the forward and backward algorithms, respectively.

It follows from equation (18) that we can apply the Viterbi algorithm with the branch metric

$$m(I_t) = \sum_{i=1}^{n_c}\sum_{j=1}^{n_c} \gamma_{t,ij}(I_{1,p}^T)\log p_{c,ij}(Y_t|X_t), \quad t=1,2,\ldots,T \tag{21}$$

to find a maximum of $Q(I_1^T, I_{1,p}^T)$ which can be interpreted as a longest path leading from the zero state to one of the states $S_T^{(s)}$. (Note that we are considering only the *encoder* trellis.) It is convenient to use the Viterbi algorithm in the backward direction to combine it with the forward-backward algorithm. In the forward direction, we compute and save in memory $\boldsymbol{\alpha}(\boldsymbol{Y}_1^t|I_{1,p}^t)$ for $t=1,2,\ldots,T$; in the backward direction, we compute recursively $\boldsymbol{\beta}(\boldsymbol{Y}_{t+1}^T|I_{t+1,p}^T)$ for $t=T-1,T-2,\ldots,1$, then we compute $\gamma_{t,ij}(I_{1,p}^T)$ and, for each encoder state at time t, determine the longest path starting from this state with branch metric $m(I_t)$ and the corresponding input sequence generating this path. Thus, the described EM algorithm can be summarized as follows:

1. Select initial sequence $I_{1,0}^T = I_{1,0}, I_{2,0}, \ldots, I_{T,0}$
2. *Forward part:*
   Set $\boldsymbol{\alpha}(\boldsymbol{Y}_1^0|I_1^0) = \boldsymbol{\pi}$ and for $t=1,2,\ldots,T$ compute $X_{t,p} = g_t(S_t^{(s)}, I_{t,p})$,
   $$\boldsymbol{\alpha}(\boldsymbol{Y}_1^t|I_{1,p}^t) = \boldsymbol{\alpha}(\boldsymbol{Y}_1^{t-1}|I_{1,p}^{t-1})\mathbf{P}_c(Y_t|X_{t,p}) \tag{22}$$
3. *Backward part:*
   Set $\boldsymbol{\beta}(\boldsymbol{Y}_{T+1}^T|I_{T+1,p}^T) = \mathbf{1}$ and the survivors [4] lengths $L(S_T^{(s)}) = 0$ for $S_T^{(s)} = 1,2,\ldots,n_s$.
   For $t=T,T-1,\ldots,1$ compute $X_t = g_t(S_t^{(s)}, I_t)$,
   $$\gamma_{t,ij}(I_{1,p}^T) = \alpha_i(\boldsymbol{Y}_1^{t-1}|I_{1,p}^{t-1})p_{c,ij}(Y_t|X_{t,p})\beta_j(\boldsymbol{Y}_{t+1}^T|I_{t+1,p}^T)$$
   $$L(S_t^{(s)}) = \max_{I_t}\{L[f_t(S_t^{(s)}, I_t)] + m(I_t)\}$$
   $$\hat{I}_t(S_t^{(s)}) = \mathop{argmax}_{I_t}\{L[f_t(S_t^{(s)}, I_t)] + m(I_t)\}$$
   $$\boldsymbol{\beta}(\boldsymbol{Y}_t^T|I_{t,p}^T) = \mathbf{P}_c(Y_t|X_{t,p})\boldsymbol{\beta}(\boldsymbol{Y}_{t+1}^T|I_{t+1,p}^T) \tag{23}$$
4. Reestimate the information sequence:
   $$I_{t,p+1} = \hat{I}_t(\hat{S}_t), \quad \hat{S}_{t+1} = f_t(\hat{S}_t, I_{t,p+1}), \quad t=1,2,\ldots,T$$
   where $\hat{S}_1 = 0$

5. If $I_{t,p+1} \neq I_{t,p}$, go to step 2; otherwise decode the information sequence as $I_{1,p+1}^T$.

Note that if the terminal state is forced to be zero, then we set $L(S_T^{(s)}) = -\infty$ for $S_T^{(s)} \neq 0$.

It is clear from the algorithm's description that the number of operations required by the algorithm is $\kappa = N_I \cdot N_{FB} \cdot N_V$ where N_{FB} is the number of operations required by the forward-backward algorithm, [1] N_V is the number of operations required by the Viterbi algorithm, [4] and N_I is the number of iterations. The number of iterations is a random number which depends on the initial approximation of the decoded sequence. Because of the discrete nature of the decoded sequence, the number of iterations is usually small. In our simulation study, for the code described in Sec. 5.2 and channel of Ref. [10], we compared the EM algorithm with the exhaustive search (for short sequences). It took less than three iterations for the algorithm to converge for various initial guesses.

The forward-backward algorithm requires saving in memory all the forward probability vectors. The backward part of the algorithm starts only after the whole sequence $\boldsymbol{Y}_1^T$ has been received. This can cause a significant problem if the sequence is long. The problem can be solved by using an approximate forward-only fixed-lag algorithm. [21] By combining the fixed-lag algorithm with the above EM algorithm it is possible to perform the decoding in the forward-only fashion. The combined algorithm lends itself to parallelization and pipelining.

This algorithm can be applied to a general IOHMM. For the model special cases the branch metric can be simplified. In particular, if the model parameters are given by equation (6), we can write [23]

$$m(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) b_j(Y_t | X_t), \quad t = 1,2,\ldots,T. \tag{24}$$

where

$$\gamma_{t,i}(I_{1,p}^T) = \alpha_i(Y_1^t | I_{1,p}^t) \beta_i(Y_{t+1}^T | I_{t+1,p}^T)$$

For the fading channel with AWGN the algorithm can be simplified. Substituting equation (8) into equation (18) we conclude that the maximization of $Q(I_1^T, I_{1,p}^T)$ is equivalent to minimization of the following quadratic form

$$R(I_1^T, I_{1,p}^T) = \sum_{t=1}^{T} \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \parallel Y_t - a_i X_t \parallel^2 \tag{25}$$

which can be accomplished by the Viterbi algorithm with the branch metric

$$m(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \parallel Y_t - a_i X_t \parallel^2, \quad t = 1,2,\ldots,T. \tag{26}$$

For a PSK modulated sequence, we can write

$$R(I_1^T, I_{1,p}^T) = -2 \sum_{t=1}^{T} \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \mathrm{Re}\{X_t Y_t^* a_i\} + C \tag{27}$$

where C is independent of I_1^T. Thus, we can use a simpler metric

$$\mu(I_t) = \sum_{i=1}^{n_c} \gamma_{t,i}(I_{1,p}^T) \mathrm{Re}\{X_t Y_t^* a_i\} \tag{28}$$

in the Viterbi algorithm.

To improve the algorithm's convergence rate, it is important to select a good initial approximation of the decoded sequence. This can be achieved by using one of the suboptimal decoding algorithms. For example, we can use the symbol MAP estimate according to equation (3), the Viterbi algorithm for the most probable path, [15] or an algebraic decoding algorithm.

To illustrate the algorithm applications, we consider two examples.

5.1 Map Decoding of Block Codes

It is well known [1,26] that block codes can be interpreted as trellis codes in the following way. Let $\mathbf{H} = [\ h_1\ h_2\ ,\ldots,\ h_N\ ]$ be a parity check matrix of a linear (N,K) code. Define the encoder states recursively as

$$S_0 = 0, \qquad S_t = S_{t-1} + I_t h_t. \tag{29}$$

Since a codeword I_1^N is in the null-space of $\mathbf{H}$ ($I_1^N \mathbf{H}' = \sum_{t=0}^{N} I_t h_t = 0$), the trellis encoder needs to keep only the trajectories leading to state $S_N = 0$. If the code is systematic, the first K symbols are the information symbols I_1^K given by the source. The parity check symbols I_{K+1}^N are uniquely defined by the path leading from state S_K to state $S_N = 0$. If the source is binary, there are only

two possible transitions from the states corresponding to information bits.

Equation (29) has the form of the first equation of (9). In block-code-based TCM, segments of a codeword X_1^N are mapped into the modulator constellation. If we have a discrete channel, the modulator is considered to be a part of the channel and the encoded symbols are transmitted directly to the channel. In this case, the second equation of (9) takes the form $X_t = I_t$. Thus, both equations (9) are satisfied and we can apply the EM decoding algorithm described in this section if the channel is modeled by an HMM.

To be more specific, consider a (5,3) block code with the parity check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The code trellis diagram is depicted in Fig. 1. This code has four states: (00), (01), (10), and (11). The encoder output bits mark the corresponding state transition edges: the (horizontal) transitions between the same states correspond to 0 and transitions between different states correspond to 1. For example, for the first observation Y_1 we have $h_1 = (1,0)'$ which is the first column of the matrix **H**.

Suppose that the channel is described by the Gilbert-Elliott model [3,10] represented by equations (6) and (7). According to these equations, we have

$$\mathbf{P}(0|0) = \mathbf{P}(1|1) = \begin{bmatrix} p_{11}(1-b_1) & p_{12}(1-b_2) \\ p_{21}(1-b_1) & p_{22}(1-b_2) \end{bmatrix}, \quad \mathbf{P}(0|1) = \mathbf{P}(1|0) = \begin{bmatrix} p_{11}b_1 & p_{12}b_2 \\ p_{21}b_1 & p_{22}b_2 \end{bmatrix}.$$

Let us assume that the sequence $Y_1^5 = 01010$ was received. As an initial approximation of the decoded sequence we choose the closest to the received sequence Y_1^5 codeword $X_{1,0}^5 = 01011$. For this codeword, we compute

$$\boldsymbol{\alpha}(Y_1|0) = \boldsymbol{\pi}_c\mathbf{P}_c(0|0), \ \boldsymbol{\alpha}(Y_1^2|01) = \boldsymbol{\alpha}(Y_1|0)\mathbf{P}_c(1|1), \ \boldsymbol{\alpha}(Y_1^3|010) = \boldsymbol{\alpha}(Y_1^2|01)\mathbf{P}_c(0|0)$$

$$\boldsymbol{\alpha}(Y_1^4|0101) = \boldsymbol{\alpha}(Y_1^3|010)\mathbf{P}_c(1|1), \quad \boldsymbol{\alpha}(Y_1^5|01011) = \boldsymbol{\alpha}(Y_1^4|0101)\mathbf{P}_c(0|1).$$

This completes the forward part. In the backward part, we compute

$$L(S_4=00) = m(X_5=0) = \gamma_{5,1}\log(1-b_1)+\gamma_{5,2}\log(1-b_2)$$

$$L(S_4=01) = m(X_5=1) = \gamma_{5,1}\log b_1+\gamma_{5,2}\log b_2$$

where $\gamma_{5,i} = \alpha_i(Y_1^5|01011)\beta_i$ and $\beta_i=1$. Then we compute $\boldsymbol{\beta}(Y_5|1) = \mathbf{P}_c(0|1)\mathbf{1}$ and

$$L(S_3=00) = m(X_4=0)+m(X_5=0),\quad L(S_3=01) = m(X_4=0)+m(X_5=1)$$

$$L(S_3=10) = m(X_4=1)+m(X_5=0),\quad L(S_3=11) = m(X_4=1)+m(X_5=1).$$

In the next step, we need to apply the Viterbi algorithm: Choose $X_3=0$ if

$$L(S_2=00) = L(S_3=00)+m(X_3=0) > L(S_3=01)+m(X_3=1).$$

Otherwise, we choose $X_3=1$ and $L(S_2=00) = L(S_3=01)+m(X_3=1)$, and so on. At the end of this iteration, we find the longest path connecting the nodes at $t=0$ and $t=5$. The corresponding encoder output sequence is the next approximation of the decoder output. The process continues until two consecutive iterations deliver the same result.

5.2 Map Decoding of Convolutional Codes

As a second example, let us consider a convolutional code. The encoder operation can be described by the following equations [5]

$$\begin{aligned} S_{t+1}^{(s)} &= S_t^{(s)}\mathbf{A} + I_t\mathbf{B} \\ \Xi_t &= S_t^{(s)}\mathbf{C} + I_t\mathbf{G} \end{aligned} \quad t=1,2,... \tag{30}$$

where S_t is the encoder state vector and Ξ_t is the encoder output vector. The encoder initial state is usually selected as $S_1 = 0$.

In the TCM standard implementation, the encoded symbols are transformed by a memoryless mapper to produce a symbol in the modulator constellation:

$$X_t = F(\Xi_t) = F(S_t^{(s)}\mathbf{C} + I_t\mathbf{G}) = f_t(S_t^{(s)},I_t). \tag{31}$$

As we can see, equations (30) and (31) have the form of equations (9). If we assume that symbols X_t are transmitted over a channel represented by equation (6), then we can apply the EM algorithm described in this section.

To be more specific, suppose that the source is binary, memoryless, and symmetric

($p(0) = p(1) = 0.5$). The source bits are coded by the rate 1/2 convolutional encoder shown in Fig. 2.

If we denote the encoder output vector as $\Xi_t = [\xi_{t1} \quad \xi_{t2}]$, then according to this figure we have

$$\begin{aligned} S_{t+1}^{(s)} &= S_t^{(s)}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + I_t[1 \quad 0] \\ \Xi_t &= S_t^{(s)}\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + I_t[1 \quad 1] \end{aligned} \tag{32}$$

where the encoder state is defined by the contents of its shift registers which is (see Fig. 2) $S_t^{(s)} = [I_{t-1}, I_{t-2}]$ and its state transition probability matrix has the form

$$\mathbf{P}_s = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix}$$

The encoder output symbols are mapped to the QPSK constellation which consists of four symbols: $X^{(0)} = (-A,-A)$, $X^{(1)} = (-A,A)$, $X^{(2)} = (A,-A)$, and $X^{(3)} = (A,A)$. Equation (16) takes the form

$$\mathbf{P}(I=0,Y)=0.5\begin{bmatrix} \mathbf{P}_c(Y|X^{(0)}) & 0 & 0 & 0 \\ \mathbf{P}_c(Y|X^{(3)}) & 0 & 0 & 0 \\ 0 & \mathbf{P}_c(Y|X^{(2)}) & 0 & 0 \\ 0 & \mathbf{P}_c(Y|X^{(1)}) & 0 & 0 \end{bmatrix} \quad \mathbf{P}(I=1,Y)=0.5\begin{bmatrix} 0 & 0 & \mathbf{P}_c(Y|X^{(3)}) & 0 \\ 0 & 0 & \mathbf{P}_c(Y|X^{(0)}) & 0 \\ 0 & 0 & 0 & \mathbf{P}_c(Y|X^{(1)}) \\ 0 & 0 & 0 & \mathbf{P}_c(Y|X^{(2)}) \end{bmatrix} \tag{33}$$

Similarly to the previous example, we can apply the EM algorithm of this section to decoding of a terminated convolutional code. If a code sequence is too long, an approximate fixed-lag algorithm can be applied in a forward-only fashion. [21] If the channel is described by equation (6), we can apply the EM algorithm whose maximization part is performed using the Viterbi algorithm with the branch metric (24).

However, if a bit-level model is described by equation (6), it does not mean that the encoded symbol model has this form. In this case we need to apply a more complex metric (21). To be more specific, suppose that the channel is described by the Gilbert-Elliott model with parameters [3,10]

$$\mathbf{P}_c = \begin{bmatrix} 0.997 & 0.003 \\ 0.034 & 0.966 \end{bmatrix}, \quad b_1 = 0.001, \quad b_2 = 0.16.$$

Thus, according to equation (6), we have

$$\mathbf{P}_c(0) = \begin{bmatrix} 0.996003 & 0.002520 \\ 0.033966 & 0.811440 \end{bmatrix}, \quad \mathbf{P}_c(1) = \begin{bmatrix} 0.000997 & 0.000480 \\ 0.000034 & 0.154560 \end{bmatrix}.$$

Suppose that the code starts and terminates at the state 00 and has three information bits. The encoder trellis diagram along with its output/input symbols are depicted in Fig. 3. For every information bit, the encoder produces a two-bit symbol which is presented in decimal form in Fig. 3. Thus, we have the following matrix probabilities of errors in the received two-bit symbols:

$$\mathbf{P}(0) = \mathbf{P}_c^2(0) = \begin{bmatrix} 0.992108 & 0.004555 \\ 0.061392 & 0.658520 \end{bmatrix}, \quad \mathbf{P}(1) = \mathbf{P}_c(0)\mathbf{P}_c(1) = \begin{bmatrix} 0.000993 & 0.000868 \\ 0.000061 & 0.125432 \end{bmatrix},$$

$$\mathbf{P}(2) = \mathbf{P}_c(1)\mathbf{P}_c(0) = \begin{bmatrix} 0.001009 & 0.000392 \\ 0.005284 & 0.125416 \end{bmatrix}, \quad \mathbf{P}(3) = \mathbf{P}_c^2(1) = \begin{bmatrix} 0.000001 & 0.000075 \\ 0.000005 & 0.023889 \end{bmatrix}.$$

These matrices do not satisfy equation (6), therefore, we cannot use the simplified metrics as in the previous examples and must use a more complex metric (21). Equation (16) takes the form

$$\mathbf{P}(I=0,Y)=0.5\begin{bmatrix} \mathbf{P}_c(Y) & 0 & 0 & 0 \\ \mathbf{P}_c(Y\oplus 3) & 0 & 0 & 0 \\ 0 & \mathbf{P}_c(Y\oplus 2) & 0 & 0 \\ 0 & \mathbf{P}_c(Y\oplus 1) & 0 & 0 \end{bmatrix} \quad \mathbf{P}(I=1,Y)=0.5\begin{bmatrix} 0 & 0 & \mathbf{P}_c(Y\oplus 3) & 0 \\ 0 & 0 & \mathbf{P}_c(Y) & 0 \\ 0 & 0 & 0 & \mathbf{P}_c(Y\oplus 1) \\ 0 & 0 & 0 & \mathbf{P}_c(Y\oplus 2) \end{bmatrix} \tag{46}$$

where $\oplus$ denotes bitwise exclusive-or operation. Suppose that we received $Y_1^5 = (0,3,2,1,0)$. As an initial guess for the transmitted codeword we choose $X_0 = 0,0,0,0,0$. Using the algorithm described on page 9, we can find a maximum of $Q(I_1^T, I_{1,p}^T)$. Obviously, it is equivalent to finding a minimum of $-Q(I_1^T, I_{1,p}^T)$ which can be achieved using the Viterbi algorithm by finding the shortest path on the trellis using negative metric in equation (21). We obtained after first iteration the following lengths of the shortest paths connecting each node of the trellis with its terminal node:

state	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
0	$0.074\cdot 10^{-3}$	$0.0618\cdot 10^{-3}$	$0.0783\cdot 10^{-3}$	$0.0443\cdot 10^{-3}$	$0.0101\cdot 10^{-3}$
1			$0.0784\cdot 10^{-3}$	$0.0442\cdot 10^{-3}$	$0.0731\cdot 10^{-3}$
2		$0.0896\cdot 10^{-3}$	$0.0511\cdot 10^{-3}$	$0.1344\cdot 10^{-3}$	
3			$0.0511\cdot 10^{-3}$	$0.0800\cdot 10^{-3}$	

The decoded sequence $\hat{I}_1^5 = 0,1,0,0,0$ and the corresponding transmitted sequence $\hat{X}_1^5 = 0,3,2,3,0$ After second iteration we obtained the following length

state	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
0	0.0248	0.0229	0.0344	0.0177	0.0020
1			0.0341	0.0171	0.0289
2		0.0382	0.0204	0.0567	
3			0.0204	0.0319	

and the same decoded sequence as in the previous iteration $\hat{I}_1^5 = 0,1,0,0,0$.

As a test, we performed an exhaustive search which gave the same result with a maximum likelihood value of 0.940201. For the HMM fading channel with the AWGN we can use metric (28).

6. CONCLUSION

We have demonstrated that if an information source, encoder, and communication channel are modeled by IOHMMs, then MAP decoding can be realized using the EM algorithm. The expectation part of the algorithm is performed using the forward-backward algorithm while the maximization part is accomplished using the Viterbi algorithm. The EM algorithm is robust and, in contrast with some other iterative algorithms, it converges to the APP maximum in all practical cases. [2] Because the set of transmitted symbols is discrete, the number of necessary iterations is usually small which was confirmed by a direct simulation.

7. ACKNOWLEDGMENT

Helpful comments and suggestions of the associate editor Dr. Steven S. Pietrobon and the anonymous reviewers are gratefully acknowledged.

References

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol IT-20, pp. 284-287, Mar. 1974.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via

the EM algorithm," *J. R. Statist. Soc.*,vol. 76, 1977, pp 341-353.

[3] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, Sept. 1963, pp. 1977-1997.

[4] G.D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE,* vol. 61, pp. 268-278, Mar. 1973.

[5] G.D. Forney, Jr., "Convolutional codes I: algebraic structure," *IEEE Trans. Inform. Theory,* vol IT-16, Nov. 1970, pp. 720-738.

[6] G.D. Forney, Jr., R.G. Gallager, G.R. Lang, F.M. Longstaff, and S.U. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 632-647, Sept. 1984.

[7] J. Garcia-Frias and J. D. Villasenor, "Exploiting binary Markov channels with unknown parameters in turbo coding," *GLOBECOM'98 Conf. Record* Sydney, 1998, pp. 3244-3249.

[8] C. N. Georghiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Comm.*, vol 45, no. 3, March, 1997.

[9] C. N. Georghiades, "Maximum-likelihood detection in the presence of interference,"*Proc. IEEE Symp. Inf. Theory*, Aug. 1998, p. 344.

[10] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, Sept. 1960, pp. 1253—1266.

[11] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory,* col IT-23, pp. 371-377, May 1977.

[12] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proc. of the IEEE,* vol. 66, no. 7, July 1978.

[13] J. H. Kang, W. E. Stark, and A. O. Hero, "Turbo codes for fading and burst channels," *GLOBECOM'98 Communications Miniconference,* Sydney, 1998, pp. 183-188.

[14] P. Ligdas, W. Turin, and N. Seshadri, "Statistical methods for speech transmission using hidden Markov models," in *Proc. 31th Conf. on Inf. Sci. and Sys.*, pp. 546-551, March 1997.

[15] N. Merhav and Y. Ephraim, "Hidden Markov modeling using a dominant state sequence with applications to speech recognition," *Computer, Speech and Language*, vol. 5, 1991, pp. 327-339.

[16] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[17] S.O. Rice, "Distribution of the duration of fades in radio transmission: Gaussian noise model," *Bell Syst. Tech Journ.*, vol. 37, May 1958, pp. 581-635.

[18] M. Sajadieh, F.R. Kschischang, and A. Leon-Garcia, "A block memory mode l for correlated Rayleigh fading channels," *Proc. IEEE Int. Conf. Commun.*, June 1996, pp. 282-286.

[19] S. Sivaprakasam and K. S. Shanmugan, "A forward-only recursion based HMM for modeling burst errors in digital channels," GLOBECOM, 1995.

[20] F. Swarts and H.C. Ferreira, "Markov characterization of channels with soft decision outputs," *IEEE Trans. Commun.*, vol. 41, May 1993, pp. 678-682.

[21] W. Turin, "MAP symbol decoding in channels with error bursts," in *Proc. 33 Conf. on Inf. Sci. and Sys.*, pp. 674-679, March 1999.

[22] W. Turin and R. van Nobelen, "Hidden Markov modeling of flat fading channels," *IEEE JSAC*, vol. 16, no. 9, Dec. 1998, pp.1809-1817.

[23] W. Turin, *Digital Transmission Systems: Performance Analysis and Modeling*, McGraw-Hill, New York, 1998.

[24] G. Ungerboeck, "Channel coding with multilevel phase signals," *IEEE Trans. Inform. Theory,* col IT-28, pp.55-67, Jan. 1982.

[25] H.S. Wang and N. Moayeri, "Finite-state Markov channel - a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, Feb. 1995, pp.163-171.

[26] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using trellis," *IEEE Trans. Inform. Theory,* vol IT-24, pp. 76-80, Jan. 1978.

[27] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, New York, pp. 213-214, 1965.

[28] L. M. Zeger and H. Kobayashi, "Sequence estimation in GSM using the EM algorithm," *Conf. on Inform. Sci. and Syst.*, March, 1998.